Bit by bit - How to fit 8 RISC-V cores in a $38 FPGA board

Olof Kindgren
Qamcom Research & Technology, FOSSi Foundation
What’s a bit-serial CPU?

A parallel CPU requires n or gates for an n-bit CPU.

0xA0 | 0x33 = 0xb3
What’s a bit-serial CPU?

$$0xA0 \mid 0x33 = 0xb3$$
What’s a bit-serial CPU?
What’s a bit-serial CPU?

A bit-serial CPU requires 1 or gate for an n-bit CPU but needs n cycles instead.

Trading speed for area
What's a bit-serial CPU?

\[ 0x03 + 0x02 = 0x05 \]
Capabilities

- RV32I with enough of privilege spec to run compliance tests and Zephyr
- Formally verified with RISCV-Formal
- Tested in hardware on TinyFPGA BX
- Packaged for FuseSoC
- BSD-licensed
Speed

● **CPI**
  ○ Most instructions take 32 cycles + memory fetch delay
  ○ Some instructions need $2 \times 32 + \text{memory fetch delay}
    ■ Jumps => Need to calculate PC and check for exceptions before updating PC
    ■ Load/store => Need to check alignment before hitting the bus
    ■ SLT* => Instructions flow LSB->MSB but bit 0 needs to be set (Grr…!!!)
  ○ Shifts need 32 + a shift-amount dependent number of cycles

● **Frequency**
  ○ ~50MHz on iCE40
  ○ ~220MHz on Artix-7
Naming conventions

SERV = CPU
Naming conventions

SERV = CPU
servant = Minimal SoC with SERV, timer, memory and GPIO. Enough to run Zephyr
Naming conventions

SERV = CPU
servant = Minimal SoC with SERV, timer, memory and GPIO. Enough to run Zephyr
servICE = servant implemented on Lattice iCE40
Naming conventions

SERV = CPU
servant = Minimal SoC with SERV, timer, memory and GPIO. Enough to run Zephyr
servICE = servant implemented on Lattice iCE40
servIVE = servant implemented on Intel Cyclone IV-E
Naming conventions

SERV = CPU
servant = Minimal SoC with SERV, timer, memory and GPIO. Enough to run Zephyr
servICE = servant implemented on Lattice iCE40
servlVE = servant implemented on Intel Cyclone IV-E
servix = servant implemented on Xilinx Artix
Naming conventions

SERV = CPU
servant = Minimal SoC with SERV, timer, memory and GPIO. Enough to run Zephyr
servICE = servant implemented on Lattice iCE40
servIVE = servant implemented on Intel Cyclone IV-E
servix = servant implemented on Xilinx Artix
servESA = servant implemented in a chip that will be used by the European Space Agency
Naming conventions

serveral = A whole lotta servants
## Size comparison

<table>
<thead>
<tr>
<th></th>
<th>TinyFPGA BX iCE40 hx8k</th>
<th>DE0 Nano Cyclone IV-E 22k</th>
<th>Digilent Nexys A7 Artix-7 100t</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERV (LUT/FF/RAM)</td>
<td>354/345/1</td>
<td>582/337/1</td>
<td>436/375/0</td>
</tr>
<tr>
<td>serveral (instances)</td>
<td>10 (out of memory)</td>
<td>33 (out of memory)</td>
<td>210 (out of memory)</td>
</tr>
<tr>
<td>$/servant</td>
<td>38/10=3.8</td>
<td>79/33=2.39</td>
<td>265/210=1.26</td>
</tr>
</tbody>
</table>

Can we do better?
More cores

serviette = Minimal SoC with CSR-less SERV, memory port and GPIO. Enough to run bare-metal RV32i
More cores

serviette = Minimal SoC with CSR-less SERV, shared memory and GPIO. Enough to run bare-metal RV32i

- serveral serviettes How many?
  - 16 to saturate memory
  - 2 seems to give best logic/memory balance
Bit by bit - How to fit 8 RISC-V cores in a $38 FPGA board

Olof Kindgren
Qamcom Research & Technology, FOSSi Foundation
Use cases

- Power management
- Glorified FSM
- Control plane CPU
- Sensor hubs
- Embarrassingly parallel tasks
Future work

● Optimizations
  ○ Share code and register file to save a RAM
  ○ Global optimization of control signals from decoder
  ○ Fine-tune decoder for LUT size (optionally put it in RAM)
  ○ General LUT golfing

● Improvements
  ○ Implement C ISA extension
  ○ Run from SPI Flash
  ○ Add RX UART and hex bootloader
That’s it folks!

https://github.com/olofk/serv
http://fusesoc.net
https://gitter.im/librecores/fusesoc

...unless you have questions

Don’t miss...